Amendments to the Specification

Please amend paragraph **[0001]** as follows:

[0001] This patent application is related to U.S. Patent Application No. 10/313,158 ~~(Attorney Docket No. 49986-0516)~~ now U.S. Pat. No. 7,171,652, filed on December 6, 2002, entitled "Software Development Environment with Design Specification Verification Tool," naming Tetsuro Motoyama and Avery Fong as inventors, which is a continuation-in-part of and claims priority to U.S. Patent Application No. 09/881,250 ~~(Attorney Docket No. 49986-0506)~~, filed on June 13, 2001, entitled "Automated Management of Development Project Files Over a Network," naming Tetsuro Motoyama as inventor. The entire contents of these prior applications are hereby incorporated by reference in their entirety for all purposes.

Please amend paragraph **[0002]** as follows:

[0002] This patent application is also related to U.S. Patent Application No. 10/059,694 ~~(Attorney Docket No. 49986-0509)~~, filed on January 28, 2002, entitled "Project Management Over A Network with Automated Task Schedule Update," naming Tetsuro Motoyama as inventor, the entire contents of which are hereby incorporated by reference in their entirety for all purposes.

Please amend paragraph **[0003]** as follows:

[0003] This patent application is also related to U.S. Patent Application No. ~~10/XXX,XXX (Attorney Docket No. 49986-0526), filed on XXX, 2003,~~ 10/652,603, filed August 28, 2003, entitled "Technique For Automating Code Generation In Developing Software Systems," naming Tetsuro Motoyama and Avery Fong as inventors.

environments) their memory is automatically managed through garbage collection.

For many programming APIs (application programming interfaces) that use reference types, it is very useful and efficient to be able to treat variables sent into them as immutable (or non-modifiable), and have the API make a copy of the variable when it is being sent into the API. The danger of not doing so is that any change to the variable that was sent into the API results in changes to its use, and those changes are often undesirable and cannot be handled appropriately. Note that value types do not have this problem, since value types are always copied on use.

However, programmers do not like the immutability requirement, and previous attempts to provide some mutable-like behavior have not been well received. By way of example, consider a graphics scene designer specifying via an API call that an object's (e.g., a button's) color is to appear red. To change the color at a later time, the designer needs to create a new object with the new color and replace the old object with it, because the old object is immutable. To this end, United States Patent Application Serial No. 10/402,268, now U.S. Patent 7,126,606 assigned to the assignee of the present invention and hereby incorporated by reference, describes a system and method in which resource objects in the system may utilize a builder

1/29/07